

Foundation of Cryptography
(0368-4162-01), Lecture 6
More on Zero Knowledge

Iftach Haitner, Tel Aviv University

December 20, 2011

Part I

Non-Interactive Zero Knowledge

Interaction is crucial for ZK

Claim 1

Assume that $\mathcal{L} \subseteq \{0, 1\}^*$ has a one-message ZK proof (even computational), with standard completeness and soundness,^a then $\mathcal{L} \in \text{BPP}$.

^aThat is, the completeness is $\frac{2}{3}$ and soundness error is $\frac{1}{3}$.

Interaction is crucial for ZK

Claim 1

Assume that $\mathcal{L} \subseteq \{0, 1\}^*$ has a one-message ZK proof (even computational), with standard completeness and soundness,^a then $\mathcal{L} \in \text{BPP}$.

^aThat is, the completeness is $\frac{2}{3}$ and soundness error is $\frac{1}{3}$.

Proof: HW

Interaction is crucial for ZK

Claim 1

Assume that $\mathcal{L} \subseteq \{0, 1\}^*$ has a one-message ZK proof (even computational), with standard completeness and soundness,^a then $\mathcal{L} \in \text{BPP}$.

^aThat is, the completeness is $\frac{2}{3}$ and soundness error is $\frac{1}{3}$.

Proof: HW

- 1 To reduce interaction we relax the zero-knowledge requirement
 - 1 Witness Indistinguishability
$$\{\langle (P(w_x^1), V^*)(x) \rangle\}_{x \in \mathcal{L}} \approx_c \{\langle (P(w_x^2), V^*)(x) \rangle\}_{x \in \mathcal{L}},$$
for any $\{w_x^1: (x, w_x) \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$ and $\{w_x^2: (x, w_x) \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$

Interaction is crucial for ZK

Claim 1

Assume that $\mathcal{L} \subseteq \{0, 1\}^*$ has a one-message ZK proof (even computational), with standard completeness and soundness,^a then $\mathcal{L} \in \text{BPP}$.

^aThat is, the completeness is $\frac{2}{3}$ and soundness error is $\frac{1}{3}$.

Proof: HW

- 1 To reduce interaction we relax the zero-knowledge requirement
 - 1 Witness Indistinguishability
$$\{\langle (P(w_x^1), V^*)(x) \rangle\}_{x \in \mathcal{L}} \approx_c \{\langle (P(w_x^2), V^*)(x) \rangle\}_{x \in \mathcal{L}},$$
for any $\{w_x^1: (x, w_x) \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$ and $\{w_x^2: (x, w_x) \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$
 - 2 Witness Hiding

Interaction is crucial for ZK

Claim 1

Assume that $\mathcal{L} \subseteq \{0, 1\}^*$ has a one-message ZK proof (even computational), with standard completeness and soundness,^a then $\mathcal{L} \in \text{BPP}$.

^aThat is, the completeness is $\frac{2}{3}$ and soundness error is $\frac{1}{3}$.

Proof: HW

- 1 To reduce interaction we relax the zero-knowledge requirement
 - 1 Witness Indistinguishability
$$\{\langle (P(w_x^1), V^*)(x) \rangle\}_{x \in \mathcal{L}} \approx_c \{\langle (P(w_x^2), V^*)(x) \rangle\}_{x \in \mathcal{L}},$$
for any $\{w_x^1: (x, w_x) \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$ and $\{w_x^2: (x, w_x) \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$
 - 2 Witness Hiding
 - 3 Non-interactive “zero knowledge”

Definition

Non-Interactive Zero Knowledge (NIZK)

Definition 2 (NIZK)

The *non interactive* PPT's (P, V) is a NIZK for $\mathcal{L} \in \text{NP}$, if $\exists p \in \text{poly}$ s.t.

Completeness: $\Pr_{c \leftarrow \{0,1\}^{p(|x|)}} [V(x, c, P(x, w_x, c)) = 1] \geq 2/3$,
for every $(x, w_x) \in R_{\mathcal{L}}$

Soundness: $\Pr_{c \leftarrow \{0,1\}^{p(|x|)}} [V(x, c, P^*(x, c)) = 1] \leq 1/3$,
for any P^* and $x \notin \mathcal{L}$

ZK: \exists PPT S s.t. $\{(x, c, P(x, w_x, c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{p(|x|)}} \approx_c \{x, S(x)\}_{x \in \mathcal{L}}$, for any $\{w_x : (x, w_x) \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$

Definition

Non-Interactive Zero Knowledge (NIZK)

Definition 2 (NIZK)

The *non interactive* PPT's (P, V) is a NIZK for $\mathcal{L} \in \text{NP}$, if $\exists p \in \text{poly}$ s.t.

Completeness: $\Pr_{c \leftarrow \{0,1\}^{p(|x|)}} [V(x, c, P(x, w_x, c)) = 1] \geq 2/3$,
for every $(x, w_x) \in R_{\mathcal{L}}$

Soundness: $\Pr_{c \leftarrow \{0,1\}^{p(|x|)}} [V(x, c, P^*(x, c)) = 1] \leq 1/3$,
for any P^* and $x \notin \mathcal{L}$

ZK: \exists PPT S s.t. $\{(x, c, P(x, w_x, c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{p(|x|)}} \approx_c \{x, S(x)\}_{x \in \mathcal{L}}$, for any $\{w_x : (x, w_x) \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$

- c – common (random) reference string (CRRS)

Definition

Non-Interactive Zero Knowledge (NIZK)

Definition 2 (NIZK)

The *non interactive* PPT's (P, V) is a NIZK for $\mathcal{L} \in \text{NP}$, if $\exists p \in \text{poly}$ s.t.

Completeness: $\Pr_{c \leftarrow \{0,1\}^{p(|x|)}} [V(x, c, P(x, w_x, c)) = 1] \geq 2/3$,
for every $(x, w_x) \in R_{\mathcal{L}}$

Soundness: $\Pr_{c \leftarrow \{0,1\}^{p(|x|)}} [V(x, c, P^*(x, c)) = 1] \leq 1/3$,
for any P^* and $x \notin \mathcal{L}$

ZK: \exists PPT S s.t. $\{(x, c, P(x, w_x, c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{p(|x|)}} \approx_c \{x, S(x)\}_{x \in \mathcal{L}}$, for any $\{w_x : (x, w_x) \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$

- c – common (random) reference string (CRRS)
- CRRS is chosen by the simulator

Definition

Non-Interactive Zero Knowledge (NIZK)

Definition 2 (NIZK)

The *non interactive* PPT's (P, V) is a NIZK for $\mathcal{L} \in \text{NP}$, if $\exists p \in \text{poly}$ s.t.

Completeness: $\Pr_{c \leftarrow \{0,1\}^{p(|x|)}} [V(x, c, P(x, w_x, c)) = 1] \geq 2/3$,
for every $(x, w_x) \in R_{\mathcal{L}}$

Soundness: $\Pr_{c \leftarrow \{0,1\}^{p(|x|)}} [V(x, c, P^*(x, c)) = 1] \leq 1/3$,
for any P^* and $x \notin \mathcal{L}$

ZK: \exists PPT S s.t. $\{(x, c, P(x, w_x, c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{p(|x|)}} \approx_c \{x, S(x)\}_{x \in \mathcal{L}}$, for any $\{w_x : (x, w_x) \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$

- c – common (random) reference string (CRRS)
- CRRS is chosen by the simulator
- What does the definition stands for?

Definition

Non-Interactive Zero Knowledge (NIZK)

Definition 2 (NIZK)

The *non interactive* PPT's (P, V) is a NIZK for $\mathcal{L} \in \text{NP}$, if $\exists p \in \text{poly}$ s.t.

Completeness: $\Pr_{c \leftarrow \{0,1\}^{p(|x|)}} [V(x, c, P(x, w_x, c)) = 1] \geq 2/3$,
for every $(x, w_x) \in R_{\mathcal{L}}$

Soundness: $\Pr_{c \leftarrow \{0,1\}^{p(|x|)}} [V(x, c, P^*(x, c)) = 1] \leq 1/3$,
for any P^* and $x \notin \mathcal{L}$

ZK: \exists PPT S s.t. $\{(x, c, P(x, w_x, c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{p(|x|)}} \approx_c \{x, S(x)\}_{x \in \mathcal{L}}$, for any $\{w_x : (x, w_x) \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$

- c – common (random) reference string (CRRS)
- CRRS is chosen by the simulator
- What does the definition stands for?
- Soundness holds also against the simulated CRRS

Definition

Non-Interactive Zero Knowledge (NIZK)

Definition 2 (NIZK)

The *non interactive* PPT's (P, V) is a NIZK for $\mathcal{L} \in \text{NP}$, if $\exists p \in \text{poly}$ s.t.

Completeness: $\Pr_{c \leftarrow \{0,1\}^{p(|x|)}} [V(x, c, P(x, w_x, c)) = 1] \geq 2/3$,
for every $(x, w_x) \in R_{\mathcal{L}}$

Soundness: $\Pr_{c \leftarrow \{0,1\}^{p(|x|)}} [V(x, c, P^*(x, c)) = 1] \leq 1/3$,
for any P^* and $x \notin \mathcal{L}$

ZK: \exists PPT S s.t. $\{(x, c, P(x, w_x, c))\}_{x \in \mathcal{L}, c \leftarrow \{0,1\}^{p(|x|)}} \approx_c \{x, S(x)\}_{x \in \mathcal{L}}$, for any $\{w_x : (x, w_x) \in R_{\mathcal{L}}(x)\}_{x \in \mathcal{L}}$

- c – common (random) reference string (CRRS)
- CRRS is chosen by the simulator
- What does the definition stands for?
- Soundness holds also against the simulated CRRS
- Amplification?

Section 1

NIZK in HBM

Hidden Bits Model (HBM)

A CRRS is chosen at random, but only the prover can see it.
The prover chooses which bits to reveal as part of the proof.

Hidden Bits Model (HBM)

A CRRS is chosen at random, but only the prover can see it. The prover chooses which bits to reveal as part of the proof. Let c^H be the “hidden” CRSS:

- Prover sees c^H , and outputs a proof π and a set on indices \mathcal{I}
- Verifier only sees the bits in c^H that are indexed by \mathcal{I}
- Simulator outputs a proof π , a set of indices \mathcal{I} and a partially hidden CRSS c^H

Hidden Bits Model (HBM)

A CRRS is chosen at random, but only the prover can see it. The prover chooses which bits to reveal as part of the proof. Let c^H be the “hidden” CRSS:

- Prover sees c^H , and outputs a proof π and a set on indices \mathcal{I}
- Verifier only sees the bits in c^H that are indexed by \mathcal{I}
- Simulator outputs a proof π , a set of indices \mathcal{I} and a partially hidden CRSS c^H

Soundness, completeness and ZK are naturally defined.

Hidden Bits Model (HBM)

A CRRS is chosen at random, but only the prover can see it. The prover chooses which bits to reveal as part of the proof. Let c^H be the “hidden” CRSS:

- Prover sees c^H , and outputs a proof π and a set on indices \mathcal{I}
- Verifier only sees the bits in c^H that are indexed by \mathcal{I}
- Simulator outputs a proof π , a set of indices \mathcal{I} and a partially hidden CRSS c^H

Soundness, completeness and ZK are naturally defined.

We give a NIZK for HC - Directed Graph Hamiltonicity, in the HBM, and then transfer it into a NIZK in the standard model.

Implies a (standard model) NIZK for all NP

Useful Matrix

- Permutation matrix: an $n \times n$ Boolean matrix, where each row/column contains a single 1

Useful Matrix

- Permutation matrix: an $n \times n$ Boolean matrix, where each row/column contains a single 1
- Hamiltonian matrix: an $n \times n$ adjacency matrix of a directed graph that consists of a single Hamiltonian cycle (note that this is also a permutation matrix)

Useful Matrix

- Permutation matrix: an $n \times n$ Boolean matrix, where each row/column contains a single 1
- Hamiltonian matrix: an $n \times n$ adjacency matrix of a directed graph that consists of a single Hamiltonian cycle (note that this is also a permutation matrix)
- An $n^3 \times n^3$ Boolean matrix is called *useful*: if it contains a generalized $n \times n$ Hamiltonian sub matrix, and all the other entries are zeros

Useful Matrix

- Permutation matrix: an $n \times n$ Boolean matrix, where each row/column contains a single 1
- Hamiltonian matrix: an $n \times n$ adjacency matrix of a directed graph that consists of a single Hamiltonian cycle (note that this is also a permutation matrix)
- An $n^3 \times n^3$ Boolean matrix is called *useful*: if it contains a generalized $n \times n$ Hamiltonian sub matrix, and all the other entries are zeros

Claim 3

Let T be a random $n^3 \times n^3$ Boolean matrix where each entry is 1 w.p n^{-5} . Hence, $\Pr[T \text{ is useful}] \in \Omega(n^{-3/2})$.

Proving Claim 3

- The expected one entries in T is $n^6 \cdot n^{-5} = n$ and by extended Chernoff bound, w.p. $\theta(1/\sqrt{n})$ T contains *exactly* n ones.

Proving Claim 3

- The expected one entries in T is $n^6 \cdot n^{-5} = n$ and by extended Chernoff bound, w.p. $\theta(1/\sqrt{n})$ T contains *exactly* n ones.
- Each row/column of T contain more than a single one entry with probability at most $\binom{n^3}{2} \cdot n^{-10} < n^{-4}$.

Proving Claim 3

- The expected one entries in T is $n^6 \cdot n^{-5} = n$ and by extended Chernoff bound, w.p. $\theta(1/\sqrt{n})$ T contains *exactly* n ones.
- Each row/column of T contain more than a single one entry with probability at most $\binom{n^3}{2} \cdot n^{-10} < n^{-4}$.
Hence, wp at least $1 - 2 \cdot n^3 \cdot n^{-4} = 1 - O(n^{-1})$, no row or column of T contains more than a single one entry.

Proving Claim 3

- The expected one entries in T is $n^6 \cdot n^{-5} = n$ and by extended Chernoff bound, w.p. $\theta(1/\sqrt{n})$ T contains *exactly* n ones.
- Each row/column of T contain more than a single one entry with probability at most $\binom{n^3}{2} \cdot n^{-10} < n^{-4}$.
Hence, wp at least $1 - 2 \cdot n^3 \cdot n^{-4} = 1 - O(n^{-1})$, no row or column of T contains more than a single one entry.
- Hence, wp $\theta(1/\sqrt{n})$ the matrix T contains a permutation matrix and all its other entries are zero.

Proving Claim 3

- The expected one entries in T is $n^6 \cdot n^{-5} = n$ and by extended Chernoff bound, w.p. $\theta(1/\sqrt{n})$ T contains *exactly* n ones.
- Each row/column of T contain more than a single one entry with probability at most $\binom{n^3}{2} \cdot n^{-10} < n^{-4}$.
Hence, wp at least $1 - 2 \cdot n^3 \cdot n^{-4} = 1 - O(n^{-1})$, no row or column of T contains more than a single one entry.
- Hence, wp $\theta(1/\sqrt{n})$ the matrix T contains a permutation matrix and all its other entries are zero.
- A random permutation matrix forms a cycle wp $1/n$ (there are $n!$ permutation matrices and $(n-1)!$ of them form a cycle)

NIZK for Hamiltonicity in HBM

NIZK for Hamiltonicity in HBM

- Common input: a directed graph $G = ([n], E)$

NIZK for Hamiltonicity in HBM

- Common input: a directed graph $G = ([n], E)$
- Common reference string T viewed as a $n^3 \times n^3$ Boolean matrix, where each entry is 1 w.p n^{-5}

NIZK for Hamiltonicity in HBM

- Common input: a directed graph $G = ([n], E)$
- Common reference string T viewed as a $n^3 \times n^3$ Boolean matrix, where each entry is 1 w.p n^{-5} ??

NIZK for Hamiltonicity in HBM

- Common input: a directed graph $G = ([n], E)$
- Common reference string T viewed as a $n^3 \times n^3$ Boolean matrix, where each entry is 1 w.p n^{-5} ??

Algorithm 4 (P)

Input: G and a cycle C in G . A CRRS $T \in \{0, 1\}_{n^3 \times n^3}$

- 1 If T not useful, set $\mathcal{I} = n^3 \times n^3$ (i.e., reveal all T) and $\phi = \perp$
Otherwise, let H be the (generalized) $n \times n$ sub matrix containing the hamiltonian cycle in T .
- 2 Set $\mathcal{I} = T \setminus H$ (i.e., , reveal the bits of T outside of H)
- 3 Choose $\phi \leftarrow \Pi_n$, s.t. C is mapped to the cycle in H
- 4 Add all the entries in H corresponding to non edges in G (with respect to ϕ) to \mathcal{I}
- 5 Output $\pi = (\mathcal{I}, \phi)$

NIZK for Hamiltonicity in HBM cont.

Algorithm 5 (V)

Input: a graph G , index set $\mathcal{I} \subseteq [n^3] \times [n^3]$, ordered set $\{T_i\}_{i \in \mathcal{I}}$ and a mapping ϕ

- ➊ If all the bits of T are revealed and T is not useful, accept. Otherwise,
- ➋ Verify that $\exists n \times n$ submatrix $H \subseteq T$ with all entries in $T \setminus H$ are zeros.
- ➌ Verify that $\phi \in \Pi_n$, and that all the entries of H not corresponding (according to ϕ) to edges of G are zeros

NIZK for Hamiltonicity in HBM cont.

Algorithm 5 (V)

Input: a graph G , index set $\mathcal{I} \subseteq [n^3] \times [n^3]$, ordered set $\{T_i\}_{i \in \mathcal{I}}$ and a mapping ϕ

- 1 If all the bits of T are revealed and T is not useful, accept. Otherwise,
- 2 Verify that $\exists n \times n$ submatrix $H \subseteq T$ with all entries in $T \setminus H$ are zeros.
- 3 Verify that $\phi \in \Pi_n$, and that all the entries of H not corresponding (according to ϕ) to edges of G are zeros

Claim 6

The above protocol is a perfect NIZK for HC in the HBM, with perfect completeness and soundness error $1 - \Omega(n^{-3/2})$

NIZK for Hamiltonicity in HBM

Proving Claim 6

- Completeness: Clear

Proving Claim 6

- Completeness: Clear
- Soundness: Assume T is useful and V accepts. Then ϕ^{-1} maps the unrevealed “edges” of H to the edges of G .

Proving Claim 6

- Completeness: Clear
- Soundness: Assume T is useful and V accepts. Then ϕ^{-1} maps the unrevealed “edges” of H to the edges of G . Hence, ϕ^{-1} maps the cycle in H to an Hamiltonian cycle in G

Proving Claim 6

- Completeness: Clear
- Soundness: Assume T is useful and V accepts. Then ϕ^{-1} maps the unrevealed “edges” of H to the edges of G . Hence, ϕ^{-1} maps the cycle in H to an Hamiltonian cycle in G
- Zero knowledge?

Algorithm 7 (S)

Input: G

- ➊ Choose T at random, according to the right distribution.
- ➋ If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$. Otherwise,
- ➌ Set $\mathcal{I} = T \setminus H$
- ➍ Let $\phi \leftarrow \Pi_n$. Replace all the entries of H not corresponding to edges of G (according to ϕ) with zeros
- ➎ Add the entries in H corresponding to non edges in G to \mathcal{I}
- ➏ Output $\pi = (T, \mathcal{I}, \phi)$

Algorithm 7 (S)

Input: G

- ➊ Choose T at random, according to the right distribution.
- ➋ If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$. Otherwise,
- ➌ Set $\mathcal{I} = T \setminus H$
- ➍ Let $\phi \leftarrow \Pi_n$. Replace all the entries of H not corresponding to edges of G (according to ϕ) with zeros
- ➎ Add the entries in H corresponding to non edges in G to \mathcal{I}
- ➏ Output $\pi = (T, \mathcal{I}, \phi)$

- Perfect simulation for non useful T 's.

Algorithm 7 (S)

Input: G

- ➊ Choose T at random, according to the right distribution.
- ➋ If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$. Otherwise,
- ➌ Set $\mathcal{I} = T \setminus H$
- ➍ Let $\phi \leftarrow \Pi_n$. Replace all the entries of H not corresponding to edges of G (according to ϕ) with zeros
- ➎ Add the entries in H corresponding to non edges in G to \mathcal{I}
- ➏ Output $\pi = (T, \mathcal{I}, \phi)$

- Perfect simulation for non useful T 's.
- For useful T , the location of H is uniform in the real and simulated case.

Algorithm 7 (S)

Input: G

- ➊ Choose T at random, according to the right distribution.
- ➋ If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$. Otherwise,
- ➌ Set $\mathcal{I} = T \setminus H$
- ➍ Let $\phi \leftarrow \Pi_n$. Replace all the entries of H not corresponding to edges of G (according to ϕ) with zeros
- ➎ Add the entries in H corresponding to non edges in G to \mathcal{I}
- ➏ Output $\pi = (T, \mathcal{I}, \phi)$

- Perfect simulation for non useful T 's.
- For useful T , the location of H is uniform in the real and simulated case.
- ϕ is a random element in Π_n in both cases

Algorithm 7 (S)

Input: G

- ➊ Choose T at random, according to the right distribution.
- ➋ If T is not useful, set $\mathcal{I} = n^3 \times n^3$ and $\phi = \perp$. Otherwise,
- ➌ Set $\mathcal{I} = T \setminus H$
- ➍ Let $\phi \leftarrow \Pi_n$. Replace all the entries of H not corresponding to edges of G (according to ϕ) with zeros
- ➎ Add the entries in H corresponding to non edges in G to \mathcal{I}
- ➏ Output $\pi = (T, \mathcal{I}, \phi)$

- Perfect simulation for non useful T 's.
- For useful T , the location of H is uniform in the real and simulated case.
- ϕ is a random element in Π_n in both cases
- Hence, the simulation is perfect

Section 2

From HBM to Standard NIZK

trapdoor permutations

Definition 8 (trapdoor permutations)

A triplet of PPT's (G, f, Inv) is called (enhanced) family of trapdoor permutation (TDP), if the following holds:

- 1 $G: \{0, 1\}^n \mapsto \{0, 1\}^n$ for every $n \in \mathbb{N}$.
- 2 $f_{pk} = f(pk, \cdot)$ is a permutation over $\{0, 1\}^n$, for every $pk \in \{0, 1\}^n$.
- 3 $\text{Inv}(sk, \cdot) \equiv f_{G(sk)}^{-1}$ for every $sk \in \{0, 1\}^n$
- 4 For any PPT A ,

$$\Pr_{x \leftarrow \{0, 1\}^n, sk \leftarrow \{0, 1\}^n, x = G(sk)} [A(pk, x) = f_{pk}^{-1}(x)] = \text{neg}(n)$$

trapdoor permutations

Definition 8 (trapdoor permutations)

A triplet of PPT's (G, f, Inv) is called (enhanced) family of trapdoor permutation (TDP), if the following holds:

- ① $G: \{0, 1\}^n \mapsto \{0, 1\}^n$ for every $n \in \mathbb{N}$.
- ② $f_{pk} = f(pk, \cdot)$ is a permutation over $\{0, 1\}^n$, for every $pk \in \{0, 1\}^n$.
- ③ $\text{Inv}(sk, \cdot) \equiv f_{G(sk)}^{-1}$ for every $sk \in \{0, 1\}^n$
- ④ For any PPT A ,

$$\Pr_{x \leftarrow \{0, 1\}^n, sk \leftarrow \{0, 1\}^n, x = G(sk)} [A(pk, x) = f_{pk}^{-1}(x)] = \text{neg}(n)$$

- For our purposes, somewhat less restrictive requirements will do

example, RSA

In the following $n \in \mathbb{N}$ and all operations are modulo n .

example, RSA

In the following $n \in \mathbb{N}$ and all operations are modulo n .

- $\mathbb{Z}_n = [n]$ and $\mathbb{Z}_n^* = \{x \in [n] : \gcd(x, n) = 1\}$

example, RSA

In the following $n \in \mathbb{N}$ and all operations are modulo n .

- $\mathbb{Z}_n = [n]$ and $\mathbb{Z}_n^* = \{x \in [n] : \gcd(x, n) = 1\}$
- $\phi(n) = |\mathbb{Z}_n^*|$ (equals $(p-1)(q-1)$ for $n = pq$ with $p, q \in \mathbb{P}$)

example, RSA

In the following $n \in \mathbb{N}$ and all operations are modulo n .

- $\mathbb{Z}_n = [n]$ and $\mathbb{Z}_n^* = \{x \in [n] : \gcd(x, n) = 1\}$
- $\phi(n) = |\mathbb{Z}_n^*|$ (equals $(p-1)(q-1)$ for $n = pq$ with $p, q \in \mathbb{P}$)
- For every $e \in \mathbb{Z}_{\phi(n)}^*$, the function $f(x) \equiv x^e$ is a permutation over \mathbb{Z}_n^* .

example, RSA

In the following $n \in \mathbb{N}$ and all operations are modulo n .

- $\mathbb{Z}_n = [n]$ and $\mathbb{Z}_n^* = \{x \in [n] : \gcd(x, n) = 1\}$
- $\phi(n) = |\mathbb{Z}_n^*|$ (equals $(p-1)(q-1)$ for $n = pq$ with $p, q \in \mathbb{P}$)
- For every $e \in \mathbb{Z}_{\phi(n)}^*$, the function $f(x) \equiv x^e$ is a permutation over \mathbb{Z}_n^* .

In particular, $(x^e)^d \equiv x \pmod{n}$, for every $x \in \mathbb{Z}_n^*$, where $d \equiv e^{-1} \pmod{\phi(n)}$

example, RSA

In the following $n \in \mathbb{N}$ and all operations are modulo n .

- $\mathbb{Z}_n = [n]$ and $\mathbb{Z}_n^* = \{x \in [n] : \gcd(x, n) = 1\}$
- $\phi(n) = |\mathbb{Z}_n^*|$ (equals $(p-1)(q-1)$ for $n = pq$ with $p, q \in \mathbb{P}$)
- For every $e \in \mathbb{Z}_{\phi(n)}^*$, the function $f(x) \equiv x^e$ is a permutation over \mathbb{Z}_n^* .

In particular, $(x^e)^d \equiv x \pmod{n}$, for every $x \in \mathbb{Z}_n^*$, where $d \equiv e^{-1} \pmod{\phi(n)}$

Definition 9 (RSA)

- $G(p, q)$ sets $pk = (n = pq, e)$ for some $e \in \mathbb{Z}_{\phi(n)}^*$, and $sk = (n, d \equiv e^{-1} \pmod{\phi(n)})$
- $f(pk, x) = x^e \pmod{n}$
- $\text{Inv}(sk, x) = x^d \pmod{n}$

example, RSA

In the following $n \in \mathbb{N}$ and all operations are modulo n .

- $\mathbb{Z}_n = [n]$ and $\mathbb{Z}_n^* = \{x \in [n]: \gcd(x, n) = 1\}$
- $\phi(n) = |\mathbb{Z}_n^*|$ (equals $(p-1)(q-1)$ for $n = pq$ with $p, q \in \mathbb{P}$)
- For every $e \in \mathbb{Z}_{\phi(n)}^*$, the function $f(x) \equiv x^e$ is a permutation over \mathbb{Z}_n^* .

In particular, $(x^e)^d \equiv x \pmod{n}$, for every $x \in \mathbb{Z}_n^*$, where $d \equiv e^{-1} \pmod{\phi(n)}$

Definition 9 (RSA)

- $G(p, q)$ sets $pk = (n = pq, e)$ for some $e \in \mathbb{Z}_{\phi(n)}^*$, and $sk = (n, d \equiv e^{-1} \pmod{\phi(n)})$
- $f(pk, x) = x^e \pmod{n}$
- $\text{Inv}(sk, x) = x^d \pmod{n}$

Factoring is easy \implies RSA is easy.

example, RSA

In the following $n \in \mathbb{N}$ and all operations are modulo n .

- $\mathbb{Z}_n = [n]$ and $\mathbb{Z}_n^* = \{x \in [n]: \gcd(x, n) = 1\}$
- $\phi(n) = |\mathbb{Z}_n^*|$ (equals $(p-1)(q-1)$ for $n = pq$ with $p, q \in \mathbb{P}$)
- For every $e \in \mathbb{Z}_{\phi(n)}^*$, the function $f(x) \equiv x^e$ is a permutation over \mathbb{Z}_n^* .

In particular, $(x^e)^d \equiv x \pmod{n}$, for every $x \in \mathbb{Z}_n^*$, where $d \equiv e^{-1} \pmod{\phi(n)}$

Definition 9 (RSA)

- $G(p, q)$ sets $pk = (n = pq, e)$ for some $e \in \mathbb{Z}_{\phi(n)}^*$, and $sk = (n, d \equiv e^{-1} \pmod{\phi(n)})$
- $f(pk, x) = x^e \pmod{n}$
- $\text{Inv}(sk, x) = x^d \pmod{n}$

Factoring is easy \implies RSA is easy. Other direction?

The transformation

The transformation

Let (P_H, V_H) be a HBM NIZK for \mathcal{L} , and let $p(n)$ be the length of the CRRS used for $x \in \{0, 1\}^n$.

Let (G, f, Inv) be a TDP and let b be an hardcore bit for f .

The transformation

Let (P_H, V_H) be a HBM NIZK for \mathcal{L} , and let $p(n)$ be the length of the CRRS used for $x \in \{0, 1\}^n$.

Let (G, f, Inv) be a TDP and let b be an hardcore bit for f .

We construct a NIZK (P, V) for \mathcal{L} , with the same completeness and “not too large” soundness error.

The transformation

The protocol

Algorithm 10 (P)

Input: $x \in \mathcal{L}$, $w \in R_{\mathcal{L}}(x)$ and CRRS $c = (c_1, \dots, c_p) \in \{0, 1\}^{np}$, where $n = |x|$ and $p = p(n)$.

- 1 Choose $sk \leftarrow U_n$, set $pk = G(sk)$ and compute $c^H = (b(z_1 = f_{pk}^{-1}(c_1)), \dots, b(z_{p(n)} = f_{pk}^{-1}(c_p)))$
- 2 Let $(\pi_H, \mathcal{I}) \leftarrow P_H(x, w, c^H)$ and output $(\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}})$

The transformation

The protocol

Algorithm 10 (P)

Input: $x \in \mathcal{L}$, $w \in R_{\mathcal{L}}(x)$ and CRRS $c = (c_1, \dots, c_p) \in \{0, 1\}^{np}$, where $n = |x|$ and $p = p(n)$.

- ① Choose $sk \leftarrow U_n$, set $pk = G(sk)$ and compute $c^H = (b(z_1 = f_{pk}^{-1}(c_1)), \dots, b(z_{p(n)} = f_{pk}^{-1}(c_p)))$
- ② Let $(\pi_H, \mathcal{I}) \leftarrow P_H(x, w, c^H)$ and output $(\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}})$

Algorithm 11 (V)

Input: $x \in \mathcal{L}$, CRRS $c = (c_1, \dots, c_p) \in \{0, 1\}^{np}$, and $(\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}})$, where $n = |x|$ and $p = p(n)$.

- ① Verify that $pk \in \{0, 1\}^n$ and that $f_{pk}(z_i) = c_i$ for every $i \in \mathcal{I}$
- ② Return $V_H(x, \pi_H, \mathcal{I}, c^H)$, where $c_i^H = b(z_i)$ for every $i \in \mathcal{I}$.

Claim 12

Assuming that (P_H, V_H) is a NIZK for \mathcal{L} in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a NIZK for \mathcal{L} with the same completeness, and soundness error α .

Claim 12

Assuming that (P_H, V_H) is a NIZK for \mathcal{L} in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a NIZK for \mathcal{L} with the same completeness, and soundness error α .

Proof: Assume for simplicity that b is unbiased (i.e., $\Pr[b(U_n) = 1] = \frac{1}{2}$).

Claim 12

Assuming that (P_H, V_H) is a NIZK for \mathcal{L} in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a NIZK for \mathcal{L} with the same completeness, and soundness error α .

Proof: Assume for simplicity that b is unbiased (i.e., $\Pr[b(U_n) = 1] = \frac{1}{2}$).

For every $pk \in \{0, 1\}^n$: $\left(b(f_{pk}^{-1}(c_1)), \dots, b(f_{pk}^{-1}(c_p)) \right)_{c \leftarrow \{0, 1\}^{np}}$ is uniformly distributed in $\{0, 1\}^p$.

Claim 12

Assuming that (P_H, V_H) is a NIZK for \mathcal{L} in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a NIZK for \mathcal{L} with the same completeness, and soundness error α .

Proof: Assume for simplicity that b is unbiased (i.e., $\Pr[b(U_n) = 1] = \frac{1}{2}$).

For every $pk \in \{0, 1\}^n$: $\left(b(f_{pk}^{-1}(c_1)), \dots, b(f_{pk}^{-1}(c_p))\right)_{c \leftarrow \{0, 1\}^{np}}$ is uniformly distributed in $\{0, 1\}^p$.

- Completeness: clear

Claim 12

Assuming that (P_H, V_H) is a NIZK for \mathcal{L} in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a NIZK for \mathcal{L} with the same completeness, and soundness error α .

Proof: Assume for simplicity that b is unbiased (i.e., $\Pr[b(U_n) = 1] = \frac{1}{2}$).

For every $pk \in \{0, 1\}^n$: $\left(b(f_{pk}^{-1}(c_1)), \dots, b(f_{pk}^{-1}(c_p))\right)_{c \leftarrow \{0, 1\}^{np}}$ is uniformly distributed in $\{0, 1\}^p$.

- Completeness: clear
- Soundness: follows by a union bound over all possible choice of $pk \in \{0, 1\}^n$.

Claim 12

Assuming that (P_H, V_H) is a NIZK for \mathcal{L} in the HBM with soundness error $2^{-n} \cdot \alpha$, then (P, V) is a NIZK for \mathcal{L} with the same completeness, and soundness error α .

Proof: Assume for simplicity that b is unbiased (i.e., $\Pr[b(U_n) = 1] = \frac{1}{2}$).

For every $pk \in \{0, 1\}^n$: $\left(b(f_{pk}^{-1}(c_1)), \dots, b(f_{pk}^{-1}(c_p))\right)_{c \leftarrow \{0, 1\}^{np}}$ is uniformly distributed in $\{0, 1\}^p$.

- Completeness: clear
- Soundness: follows by a union bound over all possible choice of $pk \in \{0, 1\}^n$.
- Zero knowledge:?

Proving zero knowledge

Algorithm 13 (S)

Input: $x \in \{0, 1\}^n$ of length n .

- Let $(\pi_H, \mathcal{I}, c^H) = S_H(x)$, where S_H is the simulator of (P_H, V_H)
- Output $M(\pi_H, \mathcal{I}, c^H) := (c, (\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}}))$, where
 - $pk \leftarrow G(U_n)$
 - Each z_i is chosen at random in $\{0, 1\}^n$ such that $b(z_i) = c_i^H$
 - $c_i = f_{pk}(z_i)$ for $i \in \mathcal{I}$, and a random value in $\{0, 1\}^n$ otherwise.

The transformation

Proving zero knowledge

Algorithm 13 (S)

Input: $x \in \{0, 1\}^n$ of length n .

- Let $(\pi_H, \mathcal{I}, c^H) = S_H(x)$, where S_H is the simulator of (P_H, V_H)
- Output $M(\pi_H, \mathcal{I}, c^H) := (c, (\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}}))$, where
 - $pk \leftarrow G(U_n)$
 - Each z_i is chosen at random in $\{0, 1\}^n$ such that $b(z_i) = c_i^H$
 - $c_i = f_{pk}(z_i)$ for $i \in \mathcal{I}$, and a random value in $\{0, 1\}^n$ otherwise.
- $M(P_H(x, w_x, C^H), C^H) \approx_c (P(x, w_x, C), C)$, where C^H and C are uniformly chosen hidden model/standart model CRS respectively

Proving zero knowledge

Algorithm 13 (S)

Input: $x \in \{0, 1\}^n$ of length n .

- Let $(\pi_H, \mathcal{I}, c^H) = S_H(x)$, where S_H is the simulator of (P_H, V_H)
- Output $M(\pi_H, \mathcal{I}, c^H) := (c, (\pi_H, \mathcal{I}, pk, \{z_i\}_{i \in \mathcal{I}}))$, where
 - $pk \leftarrow G(U_n)$
 - Each z_i is chosen at random in $\{0, 1\}^n$ such that $b(z_i) = c_i^H$
 - $c_i = f_{pk}(z_i)$ for $i \in \mathcal{I}$, and a random value in $\{0, 1\}^n$ otherwise.
- $M(P_H(x, w_x, C^H), C^H) \approx_c (P(x, w_x, C), C)$, where C^H and C are uniformly chosen hidden model/standart model CRS respectively
- Hence, distinguishing $(P(x, w_x, C), C)$ from $S(x)$ is hard

Section 3

Adaptive NIZK

Adaptive NIZK

x is chosen *after* the CRRS.

Adaptive NIZK

x is chosen *after* the CRRS.

Completeness: $\forall f: \{0, 1\}^{p(n)} \mapsto \{0, 1\}^n \cap \mathcal{L}$:
 $\Pr[V(f(c), c), P(f(c), w, c)) = 1] \geq 2/3,$

Adaptive NIZK

x is chosen *after* the CRRS.

Completeness: $\forall f: \{0, 1\}^{p(n)} \mapsto \{0, 1\}^n \cap \mathcal{L}$:

$$\Pr[V(f(c), c), P(f(c), w, c)) = 1] \geq 2/3,$$

Soundness: $\forall f: \{0, 1\}^{p(n)} \mapsto \{0, 1\}^n$ and P^*

$$\Pr[V(f(c), c, P^*((f(c), c))) = 1 \wedge f(c) \notin \mathcal{L}] \leq 1/3$$

Adaptive NIZK

x is chosen *after* the CRRS.

Completeness: $\forall f: \{0, 1\}^{p(n)} \mapsto \{0, 1\}^n \cap \mathcal{L}$:

$$\Pr[V(f(c), c), P(f(c), w, c)) = 1] \geq 2/3,$$

Soundness: $\forall f: \{0, 1\}^{p(n)} \mapsto \{0, 1\}^n$ and P^*

$$\Pr[V(f(c), c, P^*((f(c), c))) = 1 \wedge f(c) \notin \mathcal{L}] \leq 1/3$$

ZK: \exists pair of PPT's (S_1, S_2) s.t.

$\{(f(c), c, P(f(c), w_{f(c)}, c))\}_{n \in \mathbb{N}} \approx_c \{S^f(n)\}_{n \in \mathbb{N}}$, for
any $f: \{0, 1\}^{p(n)} \mapsto \{0, 1\}^n \cap \mathcal{L}$. Where $S^f(n)$ is the
output of

- 1 $(c, s) \leftarrow S_1(1^n)$
- 2 $x \leftarrow f(c)$
- 3 Output $(x, c, S_2(x, c, s))$

Adaptive NIZK

x is chosen *after* the CRRS.

Completeness: $\forall f: \{0, 1\}^{p(n)} \mapsto \{0, 1\}^n \cap \mathcal{L}$:

$$\Pr[V(f(c), c), P(f(c), w, c)) = 1] \geq 2/3,$$

Soundness: $\forall f: \{0, 1\}^{p(n)} \mapsto \{0, 1\}^n$ and P^*

$$\Pr[V(f(c), c, P^*((f(c), c))) = 1 \wedge f(c) \notin \mathcal{L}] \leq 1/3$$

ZK: \exists pair of PPT's (S_1, S_2) s.t.

$\{(f(c), c, P(f(c), w_{f(c)}, c))\}_{n \in \mathbb{N}} \approx_c \{S^f(n)\}_{n \in \mathbb{N}}$, for
any $f: \{0, 1\}^{p(n)} \mapsto \{0, 1\}^n \cap \mathcal{L}$. Where $S^f(n)$ is the
output of

- 1 $(c, s) \leftarrow S_1(1^n)$
- 2 $x \leftarrow f(c)$
- 3 Output $(x, c, S_2(x, c, s))$

- Adaptive completeness and soundness are easy to achieve from any non-adaptive NIZK.

Adaptive NIZK

x is chosen *after* the CRRS.

Completeness: $\forall f: \{0, 1\}^{p(n)} \mapsto \{0, 1\}^n \cap \mathcal{L}$:

$$\Pr[V(f(c), c), P(f(c), w, c)) = 1] \geq 2/3,$$

Soundness: $\forall f: \{0, 1\}^{p(n)} \mapsto \{0, 1\}^n$ and P^*

$$\Pr[V(f(c), c, P^*((f(c), c))) = 1 \wedge f(c) \notin \mathcal{L}] \leq 1/3$$

ZK: \exists pair of PPT's (S_1, S_2) s.t.

$\{(f(c), c, P(f(c), w_{f(c)}, c))\}_{n \in \mathbb{N}} \approx_c \{S^f(n)\}_{n \in \mathbb{N}}$, for
any $f: \{0, 1\}^{p(n)} \mapsto \{0, 1\}^n \cap \mathcal{L}$. Where $S^f(n)$ is the
output of

- 1 $(c, s) \leftarrow S_1(1^n)$
- 2 $x \leftarrow f(c)$
- 3 Output $(x, c, S_2(x, c, s))$

- Adaptive completeness and soundness are easy to achieve from any non-adaptive NIZK.
- Not every NIZK is adaptive (but the above protocol are).

Section 4

Special Soundness NIZK

Special Soundness NIZK

Part II

Proof of Knowledge

Proof of Knowledge

The protocol (P, V) is a *proof of knowledge* for $\mathcal{L} \in \text{NP}$, if P convinces V to accept x , only if it “knows” $w \in R_{\mathcal{L}}(x)$.

Proof of Knowledge

The protocol (P, V) is a *proof of knowledge* for $\mathcal{L} \in \text{NP}$, if P convinces V to accept x , only if it “knows” $w \in R_{\mathcal{L}}(x)$.

Definition 14 (knowledge extractor)

Let (P, V) be an interactive proof $\mathcal{L} \in \text{NP}$. A probabilistic machine E is a knowledge extractor for (P, V) and $R_{\mathcal{L}}$ with error $\eta: \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in \text{poly}$ s.t. $\forall x \in \mathcal{L}$ and deterministic algorithm P^* , $E^{P^*}(x)$ runs in expected time bounded by $\frac{t(|x|)}{\delta(x) - \eta(|x|)}$ and outputs $w \in R_{\mathcal{L}}(x)$, where $\delta(x) = \Pr[(P^*, V)(x) = 1]$.

Proof of Knowledge

The protocol (P, V) is a *proof of knowledge* for $\mathcal{L} \in \text{NP}$, if P convinces V to accept x , only if it “knows” $w \in R_{\mathcal{L}}(x)$.

Definition 14 (knowledge extractor)

Let (P, V) be an interactive proof $\mathcal{L} \in \text{NP}$. A probabilistic machine E is a knowledge extractor for (P, V) and $R_{\mathcal{L}}$ with error $\eta: \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in \text{poly}$ s.t. $\forall x \in \mathcal{L}$ and deterministic algorithm P^* , $E^{P^*}(x)$ runs in expected time bounded by $\frac{t(|x|)}{\delta(x) - \eta(|x|)}$ and outputs $w \in R_{\mathcal{L}}(x)$, where $\delta(x) = \Pr[(P^*, V)(x) = 1]$.

If (P, V) is a proof of knowledge (with error η), is it has a knowledge extractor with such error.

- A property of V

Proof of Knowledge

The protocol (P, V) is a *proof of knowledge* for $\mathcal{L} \in \text{NP}$, if P convinces V to accept x , only if it “knows” $w \in R_{\mathcal{L}}(x)$.

Definition 14 (knowledge extractor)

Let (P, V) be an interactive proof $\mathcal{L} \in \text{NP}$. A probabilistic machine E is a knowledge extractor for (P, V) and $R_{\mathcal{L}}$ with error $\eta: \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in \text{poly}$ s.t. $\forall x \in \mathcal{L}$ and deterministic algorithm P^* , $E^{P^*}(x)$ runs in expected time bounded by $\frac{t(|x|)}{\delta(x) - \eta(|x|)}$ and outputs $w \in R_{\mathcal{L}}(x)$, where $\delta(x) = \Pr[(P^*, V)(x) = 1]$.

If (P, V) is a proof of knowledge (with error η), is it has a knowledge extractor with such error.

- A property of V
- Why do we need it?

Proof of Knowledge

The protocol (P, V) is a *proof of knowledge* for $\mathcal{L} \in \text{NP}$, if P convinces V to accept x , only if it “knows” $w \in R_{\mathcal{L}}(x)$.

Definition 14 (knowledge extractor)

Let (P, V) be an interactive proof $\mathcal{L} \in \text{NP}$. A probabilistic machine E is a knowledge extractor for (P, V) and $R_{\mathcal{L}}$ with error $\eta: \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in \text{poly}$ s.t. $\forall x \in \mathcal{L}$ and deterministic algorithm P^* , $E^{P^*}(x)$ runs in expected time bounded by $\frac{t(|x|)}{\delta(x) - \eta(|x|)}$ and outputs $w \in R_{\mathcal{L}}(x)$, where $\delta(x) = \Pr[(P^*, V)(x) = 1]$.

If (P, V) is a proof of knowledge (with error η), is it has a knowledge extractor with such error.

- A property of V
- Why do we need it? Proving that you know the password

Proof of Knowledge

The protocol (P, V) is a *proof of knowledge* for $\mathcal{L} \in \text{NP}$, if P convinces V to accept x , only if it “knows” $w \in R_{\mathcal{L}}(x)$.

Definition 14 (knowledge extractor)

Let (P, V) be an interactive proof $\mathcal{L} \in \text{NP}$. A probabilistic machine E is a knowledge extractor for (P, V) and $R_{\mathcal{L}}$ with error $\eta: \mathbb{N} \mapsto \mathbb{R}$, if $\exists t \in \text{poly}$ s.t. $\forall x \in \mathcal{L}$ and deterministic algorithm P^* , $E^{P^*}(x)$ runs in expected time bounded by $\frac{t(|x|)}{\delta(x) - \eta(|x|)}$ and outputs $w \in R_{\mathcal{L}}(x)$, where $\delta(x) = \Pr[(P^*, V)(x) = 1]$.

If (P, V) is a proof of knowledge (with error η), is it has a knowledge extractor with such error.

- A property of V
- Why do we need it? Proving that you know the password
- Relation to ZK

Examples

Claim 15

The ZK proof we've seen in class for GI, has a knowledge extractor with error $\frac{1}{2}$.

Examples

Claim 15

The ZK proof we've seen in class for GI, has a knowledge extractor with error $\frac{1}{2}$.

Proof: ?

Examples

Claim 15

The ZK proof we've seen in class for GI, has a knowledge extractor with error $\frac{1}{2}$.

Proof: ?

Claim 16

The ZK proof we've seen in class for 3COL, has a knowledge extractor with error $\frac{1}{|E|}$.

Examples

Claim 15

The ZK proof we've seen in class for GI, has a knowledge extractor with error $\frac{1}{2}$.

Proof: ?

Claim 16

The ZK proof we've seen in class for 3COL, has a knowledge extractor with error $\frac{1}{|E|}$.

Proof: ?